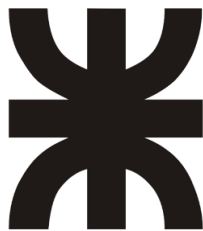


Inteligencia Artificial  
Universidad Tecnológica Nacional Regional Santa Fe

---

## TRABAJO PRÁCTICO 1 - BÚSQUEDA

---



Lomazzi, Fernando Sebastián  
Rivera, Lucio

*13 de mayo de 2024*

# 1. Ambiente

## 1.1. Estructura

Para representar al ambiente se hace uso de un grafo no dirigido en donde, a través de una etiqueta representativa, se definen los nodos y las interconexiones existentes entre sí.

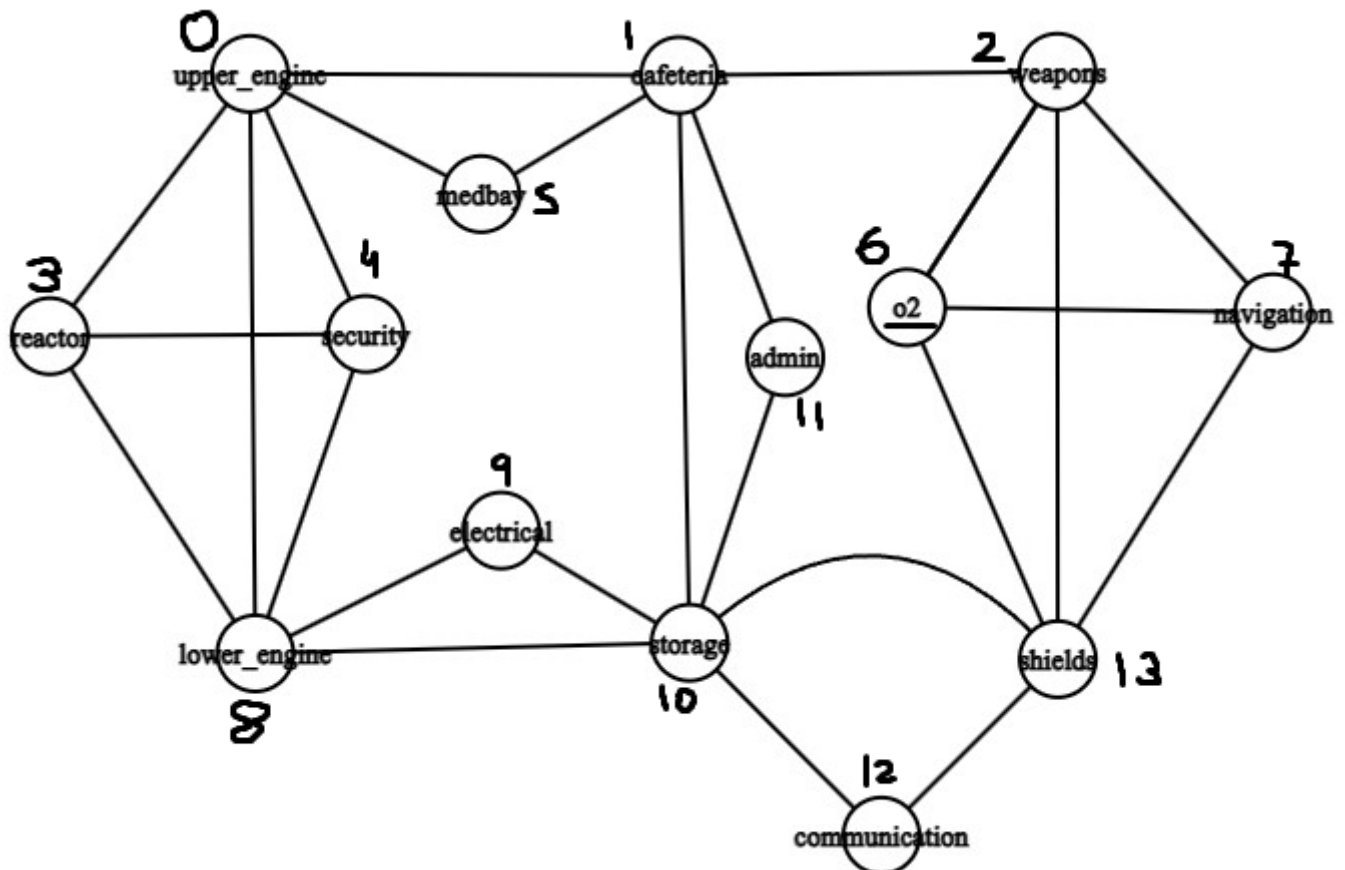


Figura 1: Representación del ambiente con el cual interactúa el agente. Consiste de catorce nodos que representan las habitaciones de la nave y sus interconexiones.

## 1.2. Estado

- Lista de Adyacencia en donde se representan las relaciones de adyacencia de las Habitaciones (nodos) de la nave.
- Lista de Tripulantes en donde cada tripulante es una estructura que contiene los siguientes elementos:
  - Habitación actual.

- Tiempo restante para próximo movimiento, en caso de valer cero, el tripulante debe moverse.
- Habitación actual en la que se encuentra el Agente.
- Tiempo restante para usar el Sensor, en caso de valer cero, el agente debe usar el sensor.

## 2. Agente

### 2.1. Estado

- Lista de relaciones de adyacencia entre las habitaciones (nodos) de la nave.
- Energía restante.
- Habitación actual.
- Lista de las habitaciones a sabotear.
- Lista de las habitaciones en las que se encuentran los tripulantes.

### 2.2. Percepciones

- Lista de habitaciones en las que se encuentran los tripulantes. En caso de no disponer el sensor, serán solo de habitaciones adyacentes a la actual.

## 3. Definición del problema

### 3.1. Espacio $\mathcal{U}$ de estados del problema

Un estado  $E \in \mathcal{U}$  del problema es representado por una 4-tupla  $(\rho, \xi, \Gamma, \Theta)$  donde cada uno de sus elementos están definidos de la siguiente forma:

- $\rho \in \text{Habitaciones}$ . Representa la habitación en la que se encuentran el agente.
- $\xi \in \mathbb{N}$ . Representa la energía restante del agente.
- $\Gamma = (\gamma_1, \gamma_2, \dots, \gamma_n) \mid \gamma_i \in \text{Habitaciones}$ . Representa la habitación en la que se encuentra el tripulante  $i$ .
- $\Theta = \{\eta \mid \eta \in \text{Habitaciones}\}$ . Representa las habitaciones que restan ser saboteadas.

## 3.2. Operadores

### 3.2.1. Moverse de la habitación $\eta_A$ a la habitación $\eta_B$ .

- Precondición:  $\eta_A$  es adyacente con  $\eta_B$  y  $\rho = \eta_A$ .
- Postcondición:  $(\eta_B, \xi - 1, \Gamma, \Theta)$ .

### 3.2.2. Eliminar al tripulante $i$ .

- Precondición:  $\rho = \gamma_i$  y  $\gamma_i \neq null$ .
- Postcondición:  $(\rho, \xi, (\dots, \gamma_{i-1}, null, \gamma_{i+1}, \dots), \Theta)$ .

### 3.2.3. Sabotear habitación $\eta$ .

- Precondición:  $\rho = \eta$  y  $\eta \in \Theta$ .
- Postcondición:  $(\rho, \xi, \Gamma, \Theta - \{\eta\})$ .

## 3.3. Prueba de meta

Sea  $\lambda : \mathcal{U} \rightarrow \{0, 1\}$  la prueba de meta asociada al problema y  $E \in \mathcal{U}$  un estado del espacio de estados, se establece que:

$$\lambda(E) = 1 \Leftrightarrow \xi > 0 \wedge \gamma_i = null \forall \gamma_i \in \Gamma \wedge \Theta = \emptyset$$

Esta última relación permite destacar dos características relevantes para la implementación del agente: en primer lugar, implica que para todos aquellos estados  $E \in \mathcal{U}$  que no satisfagan la condición del lado derecho,  $\lambda(E) = 0$ . En segundo lugar, implica la inexistencia de secuencias de operaciones que permitan alcanzar un estado objetivo para todos aquellos estados donde  $\xi = 0$ . Por este último motivo, toda implementación puede omitir la expansión de sus nodos asociados a aquellos estados donde  $\xi = 0$ .

## 3.4. Función costo y heurística propuesta

A fines de preservar la naturaleza matemática y abstraer los detalles de la implementación del problema, se introduce una función  $\Lambda(E)$ , de forma tal que  $\Lambda(E) = P$  donde  $P$  es el nodo padre de  $E$  para una estrategia de búsqueda genérica y  $E_\xi$  es el valor de  $\xi$  del estado  $E$ .

Con el auxilio de estos elementos notacionales, se propone la siguiente función costo:

$$g(E) = \begin{cases} 0 & \text{si } E \text{ es el estado inicial.} \\ g(\Lambda(E)) + \Lambda(E)_\xi - E_\xi & \text{caso contrario.} \end{cases}$$

Destáquese que esta función penalizará a aquellas acciones que consuman energía, favoreciendo así la obtención de una secuencia de acciones eficiente. Sin embargo, no

influye directamente en la velocidad de convergencia de aquellas estrategias de búsqueda que hacen uso exclusivo de esta. Bajo esta premisa, se propone la introducción de la función basada en heurísticas  $h(E)$  que dote a las estrategias de búsqueda que hagan uso de esta de una mayor velocidad de convergencia.

$$h(E) = |\Theta| + \sum_{i=1}^n \begin{cases} 1 & \gamma_i = null. \\ 0 & \text{caso contrario.} \end{cases}$$

Una hipótesis inductiva trivial permite demostrar que esta función es monótonamente decreciente y que en todo estado objetivo su valor es cero. En resumen:

$$\begin{aligned} h(E) &\leq h(\Lambda(E)) \quad \forall E \in \Omega \\ h(E) &= 0 \Leftrightarrow \lambda(E) = 1 \end{aligned}$$

Nótese que las estrategias de búsqueda informada pretenden aprovechar las virtudes tanto de la función costo como de la función basada en heurísticas, operando con estas funciones en conjunto. En el presente informe, las estrategias de búsqueda informada utilizan el siguiente criterio:

$$f(E) = g(E) + h(E)$$

## 4. Implementación

La implementación propuesta consiste de un programa ejecutable, escrito en C++ siguiendo los lineamientos del paradigma orientado a objetos que el lenguaje soporta. El sistema se basó en el siguiente modelo de clases para implementar las funcionalidades diseñadas en etapas anteriores.

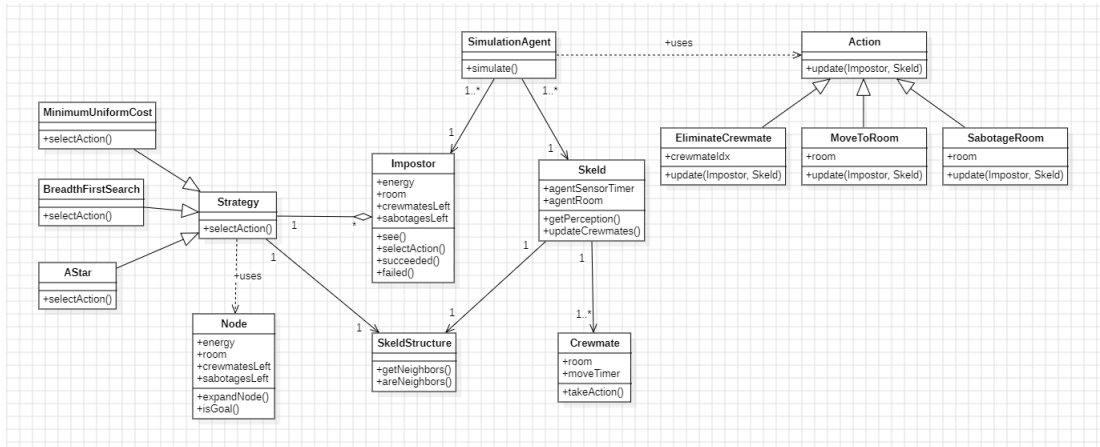


Figura 2: Diagrama de clases de la implementación propuesta

Para iniciar una simulación, primero se seleccionan parámetros como la energía, el número de tripulantes, las salas a sabotear y la estrategia de búsqueda. Estos se utilizan para crear una instancia de *SimulationAgent*. Después, se activa la simulación mediante el método *simulate*.

Durante un ciclo de la simulación, se obtiene la percepción actual del ambiente y se la envía al agente para que actualice su estado interno. Basándose en esta información, el agente decide cuál acción llevar a cabo de acuerdo con su estrategia de búsqueda. Esta acción se implementa en el agente y en el ambiente para actualizar ambos estados. Posteriormente, el estado de los tripulantes que siguen vivos es actualizado. Al final del ciclo, se verifica si la simulación ha llegado a un estado ganador o perdedor. Si no es así, el proceso se repite con el siguiente paso de la simulación.

## 4.1. Experimentos

A fines contrastar el rendimiento de las distintas estrategias de búsqueda propuestas, estas últimas fueron sometidas a distintas configuraciones iniciales en orden ascendente de dificultad. Cada configuración fue ejecutada tres veces para reducir el efecto de los errores aleatorios.

- Configuración A: 71 puntos de energía inicial y 4 tripulantes.
- Configuración B: 105 puntos de energía inicial y 8 tripulantes.
- Configuración C: 72 puntos de energía inicial y 12 tripulantes.

A continuación, se presentan los resultados obtenidos:

4 Tripulantes	Busqueda en anchura		Costo Uniforme		A-Estrella	
Semilla	Iteraciones	Tiempo Promedio	Iteraciones	Tiempo Promedio	Iteraciones	Tiempo Promedio
345	13	36.5	13	21.3	16	10
777	12	24.5	12	10.1	12	5.1
169	16	56.8	16	31.8	17	18.4
Promedio total	13.7	39.2	13.7	21.1	15.0	11.2
8 Tripulantes	Busqueda en anchura		Costo Uniforme		A-Estrella	
Semilla	Iteraciones	Tiempo Promedio	Iteraciones	Tiempo Promedio	Iteraciones	Tiempo Promedio
345	19	227.5	19	92.5	18	63.6
777	21	352.0	22	148.2	20	69.5
169	23	852.3	20	645.1	20	252.3
Promedio total	21.0	477.3	20.3	295.3	19.3	128.4
12 Tripulantes	Busqueda en anchura		Costo Uniforme		A-Estrella	
Semilla	Iteraciones	Tiempo Promedio	Iteraciones	Tiempo Promedio	Iteraciones	Tiempo Promedio
345	23	1401.0	33	426.7	23	332.3
777	27	3607.0	30	930.2	26	729.2
169	33	6407.6	28	3922.5	28	1616.5
Promedio total	27.7	3805.2	30.3	1759.8	25.7	892.7

El diseño de experimentos propuesto desvela información relevante sobre la naturaleza del problema y permite obtener una primera intuición sobre el rendimiento de cada una de las estrategias de búsqueda utilizadas.

En primer lugar, se observa que la cantidad de tripulantes iniciales es el factor inicial que más impacto tiene en la dificultad del problema subyacente. Este hecho se sustenta en la alta eficiencia que las estrategias de búsqueda obtuvieron en su totalidad, incluso en la búsqueda en anchura, que no utiliza ninguna función costo. Para ilustrar este hecho, para la tercer ejecución de la configuración C de la búsqueda en anchura, el agente necesitó de únicamente 23 percepciones, o en otras palabras, de 8 puntos de energía, para alcanzar un estado objetivo.

La diferencia significativa entre las estrategias de búsqueda propuestas está en el tiempo de ejecución de cada una de estas. El análisis de esta característica debe ser realizado sobre el promedio entre todos los tiempos.

Nótese que cada experimento está afectado por cierto grado de aleatoriedad, pero que aún así se observa una diferencia concluyente entre los tiempos de ejecución de las estrategias de búsqueda.